IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Confirmation No. 6464

Andrew G. Tucker, et al. Group Art Unit No.: 2132

Serial No.: 10/763,147 Examiner: V. Perungavoor

Filed: January 21, 2004

For: GLOBAL VISIBILITY CONTROLS FOR OPERATING SYSTEM PARTITIONS

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on February 14, 2008.

I. REAL PARTY IN INTEREST

Sun Microsystems, Inc. is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-10, 13-22, 25, and 27 are pending in the application and were finally rejected in the Final Office Action mailed on November 15, 2007. Claims 11-12, 23-24, and 26 were canceled during prosecution.

Claims 1-10, 13-22, 25, and 27 are the subject of this appeal.

IV. STATUS OF AMENDMENTS

No amendments were filed after the Final Office Action mailed on November 15, 2007.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent claims 1, 13, 25, and 27. Claim 1 is a method claim, claim 13 is a computer readable storage medium counterpart of method claim 1, claim 25 is an apparatus counterpart of method claim 1, and claim 27 is a system counterpart of method claim 1.

It has been observed by Appellants that, in some computing implementations, it is desirable to partition an operating system environment into a global zone and one or more non-global zones, where the global zone is the general operating system environment that is created when an operating system is executed and each non-global zone is a partition of the general operating system environment. Partitioned in this manner, each non-global zone may operate as a separate and distinct operating system environment within the general operating system environment. Within each non-global zone, one or more processes may be executed. A process executing within a non-global zone may be isolated within that non-global zone so that the process cannot view or access processes and objects that are not associated with that

non-global zone. By isolating processes within a non-global zone in this manner, each of the non-global zones may be made to behave like a standalone computer. This is so despite the fact that the non-global zones are part of a single operating system executing within a single computer system. It may also be desirable to allow one or more processes executing within the global zone to view and access processes and objects associated with the global zone, and processes and objects associated with the non-global zones. That way, a process in the global zone can monitor and manage both the processes and objects in the global zone and the processes and objects in the non-global zones.

Claim 1 provides a method for enabling the above desired implementation to be realized. Claim 1 recites:

> A method comprising:
> establishing a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes;
> establishing a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes;
> isolating a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone;
> permitting a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone; and
> permitting the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries.

According to the method of claim 1, a global zone is established, which can support the execution of one or more processes (see e.g. paragraphs 0035, 0038, 0068, Figs. 1, 2A, and 2B (element 130), etc.). This global zone is the general operating system environment that is created when an operating system is executed (see e.g. paragraph 0038, etc.). A non-global zone is also established (see e.g. paragraphs 0035, 0039, 0044-0049, 0068, Figs. 1,

2A, and 2B (element 140), etc.). This non-global zone is established within the global zone, and is a partition of the global operating system environment (see e.g. paragraphs 0035, 0039, 0044-0049, 0068, Figs. 1, 2A, and 2B (element 140), etc.). This non-global zone operates as a separate and distinct operating system environment, and can support execution of one or more processes (see e.g. paragraphs 0035, 0039, 0044-0049, 0068, Figs. 1, 2A, and 2B (element 140), etc.). After the non-global zone is established, a first process is executed within the non-global zone, and this first process is isolated within the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone (see e.g. paragraphs 0035, 0039, 0048, 0052, 0068, 0070, etc.). That way, the first process is prevented from viewing or accessing processes and objects in other zones. As a result, the non-global zone behaves like a standalone computer system. In addition, a second process is executed within the global zone, and this second process is permitted to have visibility and access to processes and objects associated with the global zone (see e.g. paragraphs 0035, 0052, 0069, 0072, etc.). The second process is also permitted to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries (see e.g. paragraphs 0035, 0050, 0064, 0065, 0072, etc.). By selectively permitting this access, the method of claim 1 enables a process executing within the global zone to potentially monitor and manage processes and objects associated with the non-global zone. With the method of claim 1, it is possible to implement the desired implementation discussed above.

Independent claim 13 is a computer readable storage medium counterpart of method claim 1. Thus, it is supported by at least the same portions of the Specification as those cited

above in connection with claim 1. In addition, claim 13 is supported by paragraphs 0080-0091 and Fig. 5 of the Specification. A specific mapping for claim 13 is provided below.

13. A computer readable storage medium (see e.g. paragraphs 0084, Fig. 5, etc.), comprising:

instructions for causing one or more processors to establish a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes (see e.g. paragraphs 0035, 0038, 0068, 0084, Figs. 1, 2A, and 2B (element 130), Fig. 5, etc.);

instructions for causing one or more processors to establish a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes (see e.g. paragraphs 0035, 0039, 0044-0049, 0068, 0084, Figs. 1, 2A, and 2B (element 140), Fig. 5, etc.);

instructions for causing one or more processors to isolate a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone (see e.g. paragraphs 0035, 0039, 0048, 0052, 0068, 0070, 0084, Fig. 5, etc.);

instructions for causing one or more processors to permit a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone (see e.g. paragraphs 0035, 0039, 0048, 0052, 0068, 0070, 0084, Fig. 5, etc.); and

instructions for causing one or more processors to permit the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries (see e.g. paragraphs 0035, 0050, 0064, 0065, 0072, 0084, Fig. 5, etc.).

Independent claim 25 is an apparatus counterpart of method claim 1. Thus, it is supported by at least the same portions of the Specification as those cited above in connection with claim 1. In addition, claim 25 is supported by paragraphs 0080-0091 and Fig. 5 of the Specification. A specific mapping for claim 25 is provided below.

25. An apparatus (see e.g. paragraphs 0084, Fig. 5, etc.), comprising:

means for establishing a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes (see e.g. paragraphs 0035, 0038, 0068, 0084, Figs. 1, 2A, and 2B (element 130), Fig. 5, etc.);

means for establishing a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes (see e.g. paragraphs 0035, 0039, 0044-0049, 0068, 0084, Figs. 1, 2A, and 2B (element 140), Fig. 5, etc.);

means for isolating a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone (see e.g. paragraphs 0035, 0039, 0048, 0052, 0068, 0070, 0084, Fig. 5, etc.);

means for permitting a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone (see e.g. paragraphs 0035, 0039, 0048, 0052, 0068, 0070, 0084, Fig. 5, etc.); and

means for permitting the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries (see e.g. paragraphs 0035, 0050, 0064, 0065, 0072, 0084, Fig. 5, etc.).

Independent claim 27 is a system counterpart of method claim 1. Thus, it is supported by at least the same portions of the Specification as those cited above in connection with claim 1. In addition, claim 27 is supported by paragraphs 0080-0091 and Fig. 5 of the Specification. A specific mapping for claim 27 is provided below.

A system, comprising:

one or more processors (see e.g. paragraphs 0084, Fig. 5, etc.); and

a storage (see e.g. paragraphs 0084, Fig. 5, etc.) comprising:

instructions for causing the one or more processors to establish a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes (see e.g. paragraphs 0035, 0038, 0068, 0084, Figs. 1, 2A, and 2B (element 130), Fig. 5, etc.);

instructions for causing the one or more processors to establish a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes (see e.g. paragraphs 0035, 0039, 0044-0049, 0068, 0084, Figs. 1, 2A, and 2B (element 140), Fig. 5, etc.);

instructions for causing the one or more processors to isolate a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone (see e.g. paragraphs 0035, 0039, 0048, 0052, 0068, 0070, 0084, Fig. 5, etc.);

instructions for causing the one or more processors to permit a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone (see e.g. paragraphs 0035, 0039, 0048, 0052, 0068, 0070, 0084, Fig. 5, etc.); and

instructions for causing the one or more processors to permit the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries (see e.g. paragraphs 0035, 0050, 0064, 0065, 0072, 0084, Fig. 5, etc.).

Dependent claim 4 depends from independent claim 1, and further specifies how the first process is isolated within the non-global zone. A specific mapping for claim 4 is provided below.

4. The method of claim 1, wherein the non-global zone has a first zone identifier associated therewith (see e.g. paragraph 0046, etc.), wherein processes and objects associated with the non-global zone have the first zone identifier associated therewith (see e.g. paragraphs 0073, 0074, etc.), and wherein isolating the first process to the non-global zone comprises:

allowing the first process executing within the non-global zone to view or access a target process or object only if the target process or object has the first zone identifier associated therewith (paragraphs 0073, 0074, Fig. 3G, etc.).

According to claim 4, the non-global zone has a first zone identifier associated therewith, and the processes and objects associated with the non-global zone also have the first zone identifier associated therewith (this implies that the first process executing within the non-global zone has the first zone identifier associated therewith). In such an environment, the first process is isolated within the non-global zone by allowing the first process to view or access a target process or object only if the target process or object has the first zone identifier associated therewith. In other words, the method checks the zone identifier associated with the target process or object, and allows the first process to access the target process or object only if the zone identifier of the target process or object is the same as the first zone identifier associated with the first process.

Dependent claim 16 is a computer readable storage medium counterpart of method claim 4. Thus, it is supported by at least the same portions of the Specification as those cited above in connection with claim 4. In addition, claim 16 is supported by paragraphs 0080-0091 and Fig. 5 of the Specification. A specific mapping for claim 16 is provided below.

16. The computer readable storage medium (see e.g. paragraph 0084, Fig. 5, etc.) of claim 13, wherein the non-global zone has a first zone identifier associated therewith (see e.g. paragraph 0046, etc.), wherein processes and objects associated with the non-global zone have the first zone identifier associated therewith (see e.g. paragraphs 0073, 0074, etc.), and wherein the instructions for causing one or more processors to isolate the first process to the non-global zone comprises:

instructions for causing one or more processors to allow the first process executing within the non-global zone to view or access a target process or object only if the target process or object has the first zone identifier associated therewith (paragraphs 0073, 0074, 0084, Figs. 3G and 5, etc.).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-5, 10, 13-17, 22, 25, and 27 are anticipated by Ellison et al. (U.S. Patent No. 6,663,963) under 35 U.S.C. §102(e).

2. Whether claims 6-9 and 18-21 are unpatentable over Ellison et al. (U.S. Patent No. 6,663,963) and Merkling et al. (U.S. Patent No. 5,841,869) under 35 U.S.C. §103(a).

VII. ARGUMENTS

A. The Examiner Has Erred in Rejecting Claims 1-5, 10, 13-17, 22, 25, and 27 under 35 U.S.C. §102(e)

In paragraph 4 of the Final Office Action mailed on November 15, 2007 (hereinafter, the Final Office Action), the Examiner rejected claims 1-5, 10, 13-17, 22, 25, and 27 under

35 U.S.C. §102(e) as being anticipated by Ellison et al. (U.S. Patent No. 6,663,963, hereinafter, Ellison). In order for a rejection under 35 U.S.C. §102(e) to be proper, the applied reference must show each and every limitation of the claims to which that reference is applied. In the current case, Ellison does not disclose or suggest every limitation of claims 1-5, 10, 13-17, 22, 25, and 27; thus, Ellison fails to meet this requirement. Accordingly, Appellants submit that the Examiner's rejection of claims 1-5, 10, 13-17, 22, 25, and 27 under 35 U.S.C. §102(e) is improper and, hence, request that this rejection be reversed.

Claim 1

Claim 1 recites:

A method comprising:
establishing a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes;
establishing a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes;
isolating a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone;
permitting a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone; and
permitting the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries. (Emphasis added)

Claim 1 provides an advantageous method for establishing multiple zones in an operating system environment, and for managing the visibility and accessibility of processes executing within the various zones. According to claim 1, a global zone is established, which can support the execution of one or more processes. This global zone is the global operating system environment that is created when an operating system is executed. A non-global zone

is also established. This non-global zone is established within the global zone, and is a partition of the global operating system environment (hence, the non-global zone is an operating system partition). This non-global zone operates as a separate and distinct operating system environment, and can support execution of one or more processes. After the non-global zone is established, a first process is executed within the non-global zone, and this first process is isolated within the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone. By isolating the first process in this manner, the method of claim 1 ensures that the first process is prevented from viewing or accessing processes and objects in other zones. As a result, the non-global zone behaves like a standalone computer system. In addition, a second process is executed within the global zone, and this second process is permitted to have visibility and access to processes and objects associated with the global zone. The second process is also permitted to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries. By selectively permitting this access, the method of claim 1 enables a process executing within the global zone to potentially monitor and manage processes and objects associated with the non-global zone. With the method of claim 1, it is possible to establish multiple zones in an operating system environment, and to manage the visibility and accessibility of processes executing within the various zones. Such a method is neither disclosed nor suggested by Ellison.

Instead, Ellison discloses a method, apparatus, and system for controlling memory accesses to multiple memory zones in an isolated execution environment (see Col. 2, lines 11-13). In Ellison, an accessible physical memory 60 (Fig. 1B) is divided into an isolated area 70 and a non-isolated area 80 (Col. 5, lines 1-2, Fig. 1B). The isolated area 70 is

accessible only to elements operating in isolated execution mode (Col. 5, lines 5-7). The non-isolated area 80 is accessible to all elements (Col. 5, lines 7-8). In Fig. 1A, the OS nub 16, processor nub 18, and applets 46 are shown as executing in isolated execution mode; thus, these elements can access the isolated area 70 (as shown in Fig. 1B, the OS nub 16, processor nub 18, and applets 46 can access the applet pages 72 and nub pages 74 within the isolated area 70). In Fig. 1A, the primary OS 12, software drivers 13, hardware drivers 14, and applications 42 are shown as executing in normal execution mode; thus, these elements can access the non-isolated area 80 (as shown in Fig. 1B, the primary OS 12, software drivers 13, hardware drivers 14, and applications 42 can access the application pages 82 and the OS pages 84 within the non-isolated area 80). Notice that the elements executing in normal execution mode cannot access the isolated area 70 (as shown in Fig. 1B, the primary OS 12, software drivers 13, hardware drivers 14, and applications 42 cannot access the applet pages 72 and the nub pages 74 within the isolated area 70). More importantly, notice that the elements executing in isolated execution mode can access the non-isolated area 80 (as shown in Fig. 1B, the OS nub 16, processor nub 18, and applets 46 can access the application pages 82 and the OS pages 84 in the non-isolated area 80). Ellison makes this perfectly clear in Col. 5, lines 16-20. Thus, the elements executing in isolated execution mode are not limited to accessing just the isolated area 70 but rather can access the non-isolated area 80 as well. In a way, the use of the term "isolated" is misleading because the elements executing in isolated execution mode are not isolated to the isolated area 70 at all.

The isolated area 70 may be further divided into multiple memory zones 75 (Col. 5, lines 35-37, Fig. 1C). Each memory zone 75 is associated with an operating system to define a subsystem (Col. 6, lines 6-9). A different operating system is associated with each memory

zone (Col. 6, lines 12-23), and only one memory zone and operating system is active at a time (Col. 6, line 9-10). After the various memory zones are set up, Ellison is ready to carry out the process of managing memory access. During regular operation, the process of Ellison receives access information from an access transaction (Col. 15, lines 35-36). The access information includes a physical address that is being accessed by the transaction (Col. 15, lines 36-38). The process determines whether the physical address is within the isolated memory area 70 (Col. 15, lines 38-40). If not, the process generates an access grant signal (Col. 15, lines 40-42). However, if the physical address is within the isolated memory area 70, then the process determines whether the physical address falls within the memory zone of the isolated memory area for the currently active subsystem (Col. 15, lines 43-46). If not, the process generates a failure or fault condition (Col. 15, lines 47-48). However, if the physical address does fall within the memory zone for the currently active subsystem, and if the execution mode word signal is asserted for the currently active subsystem, then the process generates an access grant signal (Col. 15, lines 48-55). By implementing this process, Ellison ensures that the access transaction is granted only if the physical address requested by a processor operating in isolated execution mode is within the correct memory zone for the currently active subsystem (Col. 15, lines 56-60).

In paragraph 4 of the Final Office Action, in rejecting claim 1, the Examiner interpreted the normal execution mode (see Fig. 1A of Ellison) and the isolated execution mode of Ellison to be the global zone recited in claim 1, and interpreted the isolated execution mode of Ellison to be the non-global zone recited in claim 1. The Examiner contended that, under this interpretation, Ellison teaches all of the limitations of claim 1. Appellants disagree and submit that the Examiner's rationale is fatally flawed.

First of all, it should be noted that the normal execution mode and the isolated execution mode of Ellison cannot in any reasonable way be interpreted to be the global and non-global zones of claim 1. The zones recited in claim 1 refer to operating system environments. The global zone is a global operating system environment and the non-global zone is a partition of the global operating system environment that operates as a separate and distinct operating system environment. Both operating system environments are capable of supporting the execution of one or more processes. In sharp contrast, the normal execution mode and isolated execution mode of Ellison are not environments; rather, they are modes of operation. This is made perfectly clear in Col. 3, lines 49-51 of Ellison, which states that the logical operating architecture 50 shown in Fig. 1A includes two modes of operation: normal execution mode and isolated execution mode. These execution modes define the mode in which a processor executes, and determine how memory accesses are managed. Specifically, if a processor is executing in isolated execution mode, then it may be allowed to access the isolated area 80 (Fig. 1B) of the physical memory 60. If the processor is executing in normal execution mode, then it is allowed to access only the non-isolated area 80 of the physical memory 60. As can be seen from the above discussion, the execution modes of Ellison are vastly different from the global and non-global zones of claim 1. Unlike the zones recited in claim 1, which are operating system environments capable of supporting the execution of one or more processes, the execution modes of Ellison are simply modes of operation that dictate which memory areas can be accessed. One has very little, if anything, in common with the other. Because they represent vastly different concepts, the execution modes of Ellison cannot be reasonably interpreted to be the global and non-global zones of claim 1.

Another point to note is that the non-global zone of claim 1 is specifically recited as a partition of the global zone. In no way can the isolated execution mode (interpreted by the Examiner to be the non-global zone of claim 1) be reasonably interpreted to be a partition of the normal execution mode and the isolated execution mode (interpreted by the Examiner to be the global zone of claim 1). The isolated execution mode is just one of two possible modes of operation: normal and isolated. Appellants do not see how the isolated execution mode is in any way a partition of the normal execution mode and the isolated execution mode. The term partition used in the context of these execution modes simply makes no sense. Thus, the isolated execution mode cannot be reasonably interpreted to be a partition of the normal execution mode and the isolated execution mode. For this additional reason, Appellants submit that the Examiner's interpretation of the normal execution mode and isolated execution mode as being the global zone of claim 1, and the isolated execution mode as being the non-global zone of claim 1, is untenable.

Given the above arguments, it is clear that the execution modes of Ellison cannot be reasonably interpreted to be the global and non-global zones of claim 1. Since Ellison does not disclose or suggest the global and non-global zones of claim 1, it follows that Ellison also does not disclose or suggest the first two limitations of claim 1, in which the global zone and the non-global zone are established.

Even if, for the sake of argument, the execution modes of Ellison could reasonably be interpreted to be the global and non-global zones of claim 1, Ellison still would not teach all of the limitations of claim 1. Particularly, claim 1 recites "isolating a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone".

No such isolation is taught by Ellison. If the isolated execution mode of Ellison is interpreted as the non-global zone of claim 1 as suggested by the Examiner, then in order for Ellison to teach the "isolating" limitation of claim 1, Ellison would have to show that the elements executing in the isolated execution mode are isolated to the isolated memory area 70 so that they do not have visibility or access to processes and objects that are not within the isolated area 70. Put another way, Ellison would have to show that the elements (such as OS nub 16, processor nub 18, and applets 46_1-46_K of Fig. 1A) that are executing in the isolated execution mode are prevented from having visibility or access to elements (such as primary OS 12, software drivers 13, hardware drivers 14, and applications 42_1-42_N) and objects that are not executing in the isolated execution mode. There is no such teaching in Ellison. In fact, Ellison teaches the opposite.

As noted previously, the elements executing in isolated execution mode are not limited to accessing just the isolated memory area 70. To the contrary, the elements executing in isolated execution mode are allowed to access the non-isolated area 80 of the physical accessible memory 60. This is made abundantly clear in numerous portions of Ellison. For example, in Col. 5, lines 16-20, Ellison specifically states that the OS nub 16 and the processor nub 18 (both of which are elements executing in isolated execution mode) can access both the isolated area 70 and the non-isolated area 80. Similarly, in Col. 6, lines 47-52, Ellison states that the OS nub 16 and the processor nub 18 can access the non-isolated area 80, including the application pages 82 and the OS pages 84, both of which are in the non-isolated area 80 (see Fig. 1B of Ellison). This excerpt further states that the applets 46 (which also execute in isolated execution mode) are likewise able to access the application pages 82 in the non-isolated area 80. From theses excerpts, and from Fig. 1B, it is

abundantly clear that the elements executing in the isolated execution mode can access the application pages 82 and the OS pages 84 in the non-isolated area 80. Notice from Fig. 1B that these application pages 82 and OS pages 84 are also accessible and usable by the elements executing in normal execution mode (e.g. primary OS 12, software drivers 13, hardware drivers 14, applications 42). Thus, these application pages 82 and OS pages 84 can, and most likely will, contain process and object data pertaining to the elements executing in normal execution mode. What this means is that the elements executing in the isolated execution mode can view and access process and object data pertaining to elements executing in the normal execution mode. This in turn means that the elements of Ellison that are executing in the isolated execution mode do have visibility and access to processes and objects that are not executing in the isolated execution mode. Thus, unlike the first process of claim 1, which is isolated to the non-global zone and hence does not have visibility or access to processes and objects that are not associated with the non-global zone, the elements of Ellison executing in the isolated execution mode are not isolated, and there is nothing in Ellison that discloses or suggests that they are isolated. In fact, as demonstrated above, Ellison teaches the opposite. That being the case, Ellison clearly does not teach or suggest the "isolating" aspect of claim 1.

As argued above, Ellison fails to teach or suggest several significant limitations of claim 1. Hence, the Examiner's rejection of claim 1 under 35 U.S.C. §102(e) based upon Ellison is improper. Accordingly, Appellants request that this rejection be reversed.

Claims 13, 25, and 27

Independent claim 13 is a computer readable storage medium counterpart of method claim 1, independent claim 25 is an apparatus counterpart of method claim 1, and independent claim 27 is a system counterpart of method claim 1. Appellants request that the rejection of these claims also be reversed for at least the reasons given above in connection with claim 1.

Claims 2-5, 10, and 14-17

Claims 2-5 and 10 depend from independent claim 1 and claims 14-17 depend from independent claim 13. Appellants request that the rejection of these claims also be reversed for at least the reasons given above in connection with claims 1 and 13.

B. The Examiner Has Erred in Rejecting Claims 4 and 16 under 35 U.S.C. §102(e)

Claim 4

Claim 4 depends from claim 1, and further recites:

> The method of claim 1, wherein the non-global zone has a first zone identifier associated therewith, wherein processes and objects associated with the non-global zone have the first zone identifier associated therewith, and wherein isolating the first process to the non-global zone comprises:
> allowing the first process executing within the non-global zone to view or access a target process or object only if the target process or object has the first zone identifier associated therewith.

Claim 4 elaborates upon what is performed to isolate the first process to the non-global zone. According to claim 4, the non-global zone has a first zone identifier associated therewith, and the processes and objects associated with the non-global zone also have the first zone identifier associated therewith (this implies that the first process executing within

the non-global zone has the first zone identifier associated therewith). In such an environment, the first process is isolated within the non-global zone by allowing the first process to view or access a target process or object only if the target process or object has the first zone identifier associated therewith. In other words, the method checks the zone identifier associated with the target process or object, and allows the first process to access the target process or object only if the zone identifier of the target process or object is the same as the first zone identifier associated with the first process. Such a method is neither disclosed nor suggested by Ellison.

As argued above in connection with claim 1, Ellison does not disclose or suggest the "isolating" limitation of claim 1. Ellison certainly does not disclose or suggest the elaboration on the "isolating" limitation set forth in claim 4. Specifically, there is nothing in Ellison that discloses or suggests that the isolated execution mode (interpreted by the Examiner as the non-global zone in the claims) has an identifier associated therewith. There is also nothing in Ellison that discloses or suggests that the elements executing in the isolated execution mode have this identifier associated therewith. Furthermore, there is nothing in Ellison that discloses or suggests that an element executing in the isolated execution mode is allowed to view or access a target process or object only if the target process or object has this identifier associated therewith. Thus, none of the aspects set forth in claim 4 are disclosed or suggested by Ellison. In support of the rejection, the Examiner, in paragraph 6 of the Final Office Action, referred to the identifiers associated with the rings, applications, and applets shown in Fig. 1A. However, none of these identifiers are associated with the isolated execution mode (which has been interpreted by the Examiner as the non-global zone), and none of these identifiers are used to determine whether an element executing in

the isolated execution mode is allowed to view or access a target process or object. Thus, Ellison does not disclose or suggest the aspects set forth in claim 4. Hence, the Examiner's rejection of claim 4 under 35 U.S.C. §102(e) based upon Ellison is improper. Accordingly, Appellants request that this rejection be reversed.

Claim 16

Claim 16 is a computer readable storage medium counterpart of method claim 4. Appellants request that the rejection of this claim also be reversed for at least the reasons given above in connection with claim 4.

C. The Examiner Has Erred in Rejecting Claims 6-9 and 18-21 under 35 U.S.C. §103(a)

In paragraph 8 of the Final Office Action, the Examiner rejected claims 6-9 and 18-21 under 35 U.S.C. §103(a) as being unpatentable over Ellison in view of Merkling et al. (U.S. Patent No. 5,841,869, hereinafter, Merkling). In order for a rejection under 35 U.S.C. §103(a) to be proper, the applied references, when properly combined, must show each and every limitation of a claim. The Examiner has failed to meet this requirement. Specifically, even when combined (assuming for the sake of argument that it would have been obvious to combine the references) Ellison and Merkling do not show each and every limitation of claims 6-9 and 18-21. Thus, Appellants submit that this rejection is improper, and request that it be reversed.

Claims 6-9

Claims 6-9 depend from claim 1; thus, they incorporate all of the limitations of claim 1. As argued above in connection with claim 1, Ellison fails to disclose or suggest several aspects of claim 1. These same aspects of claim 1 are also not disclosed or suggested by Merkling. In fact, the Examiner has made no allegation that Merkling teaches these aspects of claim 1. Since neither reference discloses or suggests at least these aspects of claim 1, even if the references were combined (assuming for the sake of argument that it would have been obvious to combine the references), the combination still would not yield the invention as claimed in claim 1. Thus, Ellison and Merkling fail to disclose or suggest all of the limitations of claim 1. Ellison and Merkling also fail to disclose or suggest all of the limitations of claims 6-9. Accordingly, Appellants submit that the rejection of claims 6-9 under 35 U.S.C. §103(a) based upon Ellison and Merkling is improper, and request that this rejection be reversed.

Claims 18-21

Claims 18-21 are computer readable storage medium counterparts of method claims 6-9. Appellants request that the rejection of these claims also be reversed for at least the reasons given above in connection with claims 6-9.

D. Conclusion and Prayer for Relief

Based on the foregoing, it is respectfully submitted that the rejections of claims 1-10, 13-22, 25, and 27 are improper and lack the requisite factual and legal bases. Therefore, Appellants respectfully request that the Honorable Board reverse the rejections of claims 1-10, 13-22, 25, and 27, and hold the claims to be allowable.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

/BobbyKTruong#37499/
Bobby Truong
Reg. No. 37,499
Date: May 1, 2008

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Tel: (408) 414-1080 ext. 234
Fax: (408) 414-1076

VIII. Claims Appendix

1. (Previously Presented) A method comprising:

establishing a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes;

establishing a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes;

isolating a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone;

permitting a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone; and

permitting the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries.

2. (Previously Presented) The method of claim 1, further comprising:

permitting the second process executing within the global zone to have visibility of processes and objects associated with the non-global zone without requiring the second process to have the privilege to cross zone boundaries.

3. (Previously Presented) The method of claim 1, further comprising:

receiving a request from the second process executing within the global zone to cross zone boundaries; and

granting the second process the privilege to cross zone boundaries, if the second process is authorized to receive such a privilege.

4. (Previously Presented) The method of claim 1, wherein the non-global zone has a first zone identifier associated therewith, wherein processes and objects associated with the non-global zone have the first zone identifier associated therewith, and wherein isolating the first process to the non-global zone comprises:

allowing the first process executing within the non-global zone to view or access a target process or object only if the target process or object has the first zone identifier associated therewith.

5. (Previously Presented) The method of claim 4, wherein the global zone has a second zone identifier associated therewith, wherein processes and objects associated with the global zone have the second zone identifier associated therewith, and wherein permitting the second process to have visibility and access to processes and objects associated with the global zone comprises:

allowing the second process executing within the global zone to view and access an intended process or object if the intended process or object has the second zone identifier associated therewith.

6. (Previously Presented) The method of claim 1, further comprising:

receiving an identifier indicating a zone selected from at least one of the global zone and the non-global zone; and

mounting file system resources comprising processes to be executed in the zone indicated by the identifier to a portion of a file system associated with the zone indicated by the identifier;

thereby enabling the processes of the file system resources to obtain at least one of visibility and access to objects within the zone corresponding to the identifier.

7. (Original) The method of claim 6, wherein the file system resources are mounted to a subdirectory of a root directory of a portion of a file system associated with the zone indicated by the identifier; thereby enabling processes expecting a tree like directory structure to execute within the zone indicated by the identifier.

8. (Original) The method of claim 6, further comprising:

enabling select processes to be visible to all other processes in the global zone and the non-global zone.

9. (Previously Presented) The method of claim 6, wherein file system resources comprise processes to be executed in any zone, the method further comprising:

receiving a request by a requesting process to access processes in the file system resources; and

limiting access to processes in the file system resources based upon the requesting process' relationship with a zone indicated in the request;

thereby enabling the processes of the file system resources to obtain at least one of visibility and access to objects within the zone corresponding to the identifier.

10. (Original) The method of claim 1, further comprising:

providing information about the zone with which a process is associated based upon identity of a requesting process and relationship between the requesting process and the zone.

11. Canceled

12. Canceled

13. (Previously Presented) A computer readable storage medium, comprising:

instructions for causing one or more processors to establish a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes;

instructions for causing one or more processors to establish a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes;

instructions for causing one or more processors to isolate a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone;

instructions for causing one or more processors to permit a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone; and

instructions for causing one or more processors to permit the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries.

14. (Previously Presented) The computer readable storage medium of claim 13, further comprising:

instructions for causing one or more processors to permit the second process executing within the global zone to have visibility of processes and objects associated with the non-global zone without requiring the second process to have the privilege to cross zone boundaries.

15. (Previously Presented) The computer readable storage medium of claim 13, further comprising:

instructions for causing one or more processors to receive a request from the second process executing within the global zone to cross zone boundaries; and

granting the second process the privilege to cross zone boundaries, if the second process is authorized to receive such a privilege.

16. (Previously Presented) The computer readable storage medium of claim 13, wherein the non-global zone has a first zone identifier associated therewith, wherein processes and objects associated with the non-global zone have the first zone identifier associated therewith, and wherein the instructions for causing one or more processors to isolate the first process to the non-global zone comprises:

instructions for causing one or more processors to allow the first process executing within the non-global zone to view or access a target process or object only if the target process or object has the first zone identifier associated therewith.

17. (Previously Presented) The computer readable storage medium of claim 16, wherein the global zone has a second zone identifier associated therewith, wherein processes and objects associated with the global zone have the second zone identifier associated therewith, and wherein the instructions for causing one or more processors to permit the second process to have visibility and access to processes and objects associated with the global zone comprises:

instructions for causing one or more processors to allow the second process executing within the global zone to view and access an intended process or object if the intended process or object has the second zone identifier associated therewith.

18. (Previously Presented) The computer readable storage medium of claim 13, further comprising:

instructions for causing one or more processors to receive an identifier indicating a zone selected from at least one of the global zone and the non-global zone; and

instructions for causing one or more processors to mount file system resources comprising processes to be executed in the zone indicated by the identifier to a portion of a file system associated with the zone indicated by the identifier.

19. (Previously Presented) The computer readable storage medium of claim 18, wherein the file system resources are mounted to a subdirectory of a root directory of a portion of a file system associated with the zone indicated by the identifier;

thereby enabling processes expecting a tree like directory structure to execute within the zone indicated by the identifier.

20. (Previously Presented) The computer readable storage medium of claim 18, further comprising:

instructions for causing one or more processors to enable select processes to be visible to all other processes in the global zone and the non-global zone.

21. (Previously Presented) The computer readable storage medium of claim 18, wherein file system resources comprise processes to be executed in any zone, and wherein the computer readable storage medium further comprises:

instructions for causing one or more processors to receive a request by a requesting process to access processes in the file system resources; and

instructions for causing one or more processors to limit access to processes in the file system resources based upon a requesting process' relationship with a zone indicated in the request.

22. (Previously Presented) The computer readable storage medium of claim 13, further comprising:

instructions for causing one or more processors to provide information about the zone with which a process is associated based upon identity of a requesting process and relationship between the requesting process and the zone.

23. Canceled

24. Canceled

25. (Previously Presented) An apparatus, comprising:

means for establishing a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes;

means for establishing a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global zone operates as a separate and distinct operating system

environment, and wherein the non-global zone can support execution of one or more processes;

means for isolating a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone;

means for permitting a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone; and

means for permitting the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries.

26. Canceled

27. (Previously Presented) A system, comprising:

one or more processors; and

a storage comprising:

instructions for causing the one or more processors to establish a global zone, wherein the global zone is a global operating system environment that can support execution of one or more processes;

instructions for causing the one or more processors to establish a non-global zone within the global zone, wherein the non-global zone is a partition of the global operating system environment, wherein the non-global

zone operates as a separate and distinct operating system environment, and wherein the non-global zone can support execution of one or more processes;

instructions for causing the one or more processors to isolate a first process executing within the non-global zone to the non-global zone so that the first process does not have visibility or access to processes and objects that are not associated with the non-global zone;

instructions for causing the one or more processors to permit a second process executing within the global zone to have visibility and access to processes and objects associated with the global zone; and

instructions for causing the one or more processors to permit the second process executing within the global zone to have access to processes and objects associated with the non-global zone, if the second process has a privilege to cross zone boundaries.

IX. Evidence Appendix

None

X. Related Proceedings Appendix

None